



Microsoft

70-526-CSharp

Microsoft .NET Framework 2.0 -(R) Windows-based Client Development is in development (Csharp)

QUESTION: 68

You want to execute an event handler asynchronously from a Windows Form. You need to write code that uses the BackgroundWorker component named bgwExecute to execute the WorkHandler method. Which code segment should you use?

- A. `EventHandler work = new EventHandler(WorkHandler);
bgwExecute.RunWorkerAsync(work);`
- B. `ThreadStart tsBackground = new ThreadStart(WorkHandler);
bgwExecute.ReportProgress(100, tsBackground);`
- C. `ThreadStart tsBackground = new ThreadStart(WorkHandler);
bgwExecute.RunWorkerAsync(tsBackground);`
- D. `bgwExecute.DoWork += new DoWorkEventHandler(WorkHandler);
bgwExecute.RunWorkerAsync();`

Answer: D

QUESTION: 69

You are customizing a Windows Form. The form includes a menu that has several ToolStripMenuItem controls. An event handler is configured to handle the Click event for all ToolStripMenuItem controls. The event handler has the following signature.

```
private void menu_Click(object sender, EventArgs e)
```

The form class includes a method that has the following signature.

```
private void LogClick(string ctlName)
```

You need to add code so that when a user clicks a ToolStripMenuItem control, the menu_Click method calls the LogClick method. The LogClick method must be called with the ctlName parameter set to the menu text in the ToolStripMenuItem control. Which code segment should you use?

- A. `ToolStripMenuItem mnuItem = (ToolStripMenuItem)sender;
LogClick(mnuItem.Text);`
- B. `LogClick(e.ToString());`
- C. `LogClick(this.Text);`
- D. `ToolStripMenuItem mnuItem = (ToolStripMenuItem) this.GetContainerControl();
LogClick(mnuItem.Text);`

Answer: A

QUESTION: 70

You are creating a Windows Form that includes a TextBox control named txtDate. When a user right click within the text box, you want the application to display a

MonthCalendar control. You need to implement a context menu that provides this functionality. What should you do?

A. Add the following code to the form initialization.

```
MonthCalendar cal = new MonthCalendar();
ContextMenuStrip mnuContext = new ContextMenuStrip();
ToolStripControlHost host = new ToolStripControlHost(mnuContext);
txtDate.ContextMenuStrip = mnuContext;
```

B. Add the following code to the form initialization.

```
ContextMenuStrip mnuContext = new ContextMenuStrip();
MonthCalendar cal = new MonthCalendar();
```

```
ToolStripControlHost host = new
    ToolStripControlHost(cal); mnuContext.Items.Add(host);
```

```
txtDate.ContextMenuStrip = mnuContext;
```

C. Add the following code to the form initialization.

```
ToolStripContainer ctr = new ToolStripContainer();
MonthCalendar cal = new MonthCalendar();
ctr.ContentPanle.Controls.Add(cal); txtDate.Controls.Add(ctr);
```

Add a MouseClick event handler for the TextBox control that contains the following code. if (e.Button == MouseButton.Right) {
txtDate.Controls[0].Show();
}

D. Add a MouseClick event handler for the TextBox control that contains the following code. if (e.Button == MouseButtons.Right) {

```
ContextMenuStrip mnuContext = new ContextMenuStrip();
MonthCalendar cal = new MonthCalendar();
ToolStripControlHost host = new
```

Answer: B

QUESTION: 71

You are creating a Windows Forms application. The application uses a SqlCommand object named cmd. The cmd object executes the following stored procedure.

```
CREATE PROCEDURE GetPhoneList
AS
BEGIN
    SELECT CompnayName, Phone FROM Customers
    SELECT CompanyName, Phone FROM Suppliers
END
```

You need to add all returned rows to the ListBox control named lstPhones. Which code segment should you use?

A. SqlDataReader rdr = cmd.ExecuteReader();

```

do {
    while (rdr.Read()) {
        lstPhones.Items.Add(rdr.GetString(0) + "\t"
            + rdr.GetString(1));
    }
} while (rdr.NextResult());
B. SqlDataReader rdr = cmd.ExecuteReader();
while (rdr.Read()) {
    lstPhones.Items.Add(rdr.GetString(0) + "\t"
        + rdr.GetString(1));
}
C. SqlDataReader rdr = cmd.ExecuteReader();
while (rdr.NextResult()) {
    while (rdr.Read()) {
        lstPhones.Items.Add(rdr.GetString(0) + "\t"
            + rdr.GetString(1));
    }
}
D. SqlDataReader rdr = cmd.ExecuteReader();
while (rdr.NextResult()) {
    lstPhones.Items.Add(rdr.GetString(0) + "\t"
        + rdr.GetString(1));
}

```

Answer: A

QUESTION: 72

A Windows Forms application reads the following XML file.

```

<?xml version="1.0"?>
<x:catalog xmlns:x="urn:books">
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
  </book>
  <book id="bk102">
    <author>Ralis, Kim</author>
    <title>Midnight Rain</title>
  </book>
</x:catalog>

```

The form initialization loads this file into an XmlDocument object named docBooks. You need to populate a ListBox control named lstBooks with the concatenated book ID and title of each book. Which code segment should you use?

```

A. XmlNodeList elements = docBooks.GetElementsByTagName("book");
foreach (XmlElement node in elements) {
    string s = node.GetAttribute("id") + " - ";
    s += node.SelectSingleNode("title").InnerText;
    lstBooks.Items.Add(s);
}
B. XmlNodeList elements = docBooks.GetElementsByTagName("book");
foreach (XmlElement node in elements) {
    string s = node.SelectSingleNode("id") + " - ";
    s += node.GetAttribute("title");
    lstBooks.Items.Add(s);
}
C. XmlNodeList elements = docBooks.GetElementsByTagName("book");
foreach (XmlElement node in elements) {
    string s = node.GetAttribute("id") + " - ";
    s += node.SelectSingleNode("title").Value;
    lstBooks.Items.Add(s);
}
D. XmlNodeList elements = docBooks.GetElementByTagName("book");
foreach (XmlElement node in elements) {
    lstBooks.Items.Add(node.InnerXml);
}

```

Answer: A

QUESTION: 73

A Windows Forms application contains the following code segment.

```

tring SQL = @"SELECT EmployeeID, LastName, FirstName, FROM
Employees";
SqlDataAdapter da = new SqlDataAdapter(SQL, connStr);
DataTable dt = new DataTable();
da.MissingSchemaAction = MissingSchemaAction.AddWithKey; SqlCommandBuilder
bld = new SqlCommandBuilder(da); da.Fill(dt);

```

The application allows the user to add rows to the data table. The application will propagate these additions to the database. If the addition of any row fails, the other rows must still be added. The code must log how many new rows to be added. You need to propagate the additions to the database and log a failed count. Which code segment should you use?

```

A. da.ContinueUpdateError = true;
da.Update(dt);
DataTable dtErrors = dt.GetChanges(DataRowState.Unchanged);
Trace.WriteLine(dtErrors.Rows.Count.ToString() +

```

```

" rows not added.");
B. da.ContinueUpdateOnError = false;
da.Update(dt);
DataTable dtErrors = dt.GetChanges(DataRowState.Unchanged);
Trace.WriteLine(dtErrors.Rows.Count.ToString() +
" rows not added.");
C. da.ContinueUpdateOnError = true;
da.Update(dt);
DataRow[] rows = dt.GetErrors(); Trace.WriteLine(rows.Length.ToString() + " rows not
added.");
D. da.ContinueUpdateOnError = false;
da.Update(dt);
DataRow[] rows = dt.GetErrors(); Trace.WriteLine(rows.Length.ToString() + " rows not
added.");

```

Answer: C

QUESTION: 74

You are customizing a Windows Form to asynchronously update a database. You need to ensure that the form display a message box to the user that indicates the success or failure of the update. Which three code segment should you use? (Each correct answer presents part of the solution. Choose three.)

```

A. private void StartBackGroundProcess() {
    bgwExecute.DoWork += new DoWorkEventHandler(WorkHandler);
    bgwExecute.RunWorkerCompleted +=
    new RunWorkerCompletedEventHandler(CompletedHandler);
    bgwExecute.RunWorkerAsync();
}
B. private void StartBackgroundProcess() {
    bgwExecute.ProcessChanged +=
    new ProgressChangedEventHandler(CompletedHandler);
    ThreadStart tsBackground = new ThreadStart(WorkHandler);
    bgwExecute.RunWorkerAsync(tsBackground);
}
C. private void StartBackgroundProcess() {
    bgwExecute.RunWorkerCompleted +=
    new RunWorkerCompletedEventHandler(CompletedHandler);
    ThreadStart tsBackground = new ThreadStart(WorkHandler);
    bgwExecute.RunWorkerAsync(tsBackground);
}
D. void WorkHandler(object sender, DoWorkEventArgs e) {
    // ...

```

```

E. Result = true;
}
F. void WorkHandler(object sender, DoWorkEventArgs e) {
// ...
bgwExecute.ReportProgress(100, true);
}
G. void CompletedHandler(object sender, RunWorkerCompletedEventArgs e) {
bool result = (bool)e.Result;
MessageBox.Show("Update " + (result ? "was successful" : "failed"));
}
H. void ProgressHandler(object sender, ProgressChangedEventArgs e)
{
bool result = (bool)e.UserState;
MessageBox.Show("Update " + (result ? "was successful" : "failed"));
}

```

Answer: A,D,G

QUESTION: 75

You are customizing a Windows Form to update a database asynchronously by using an instance of a BackgroundWorker component named bgwExecute. You start the component by using the following code.

```

private void StartBackgroundProcess() {
bgwExecute.DoWork += new DoWorkEventHandler(WorkHandler);
bgwExecute.DoWorkerCompleted +=
new DoWorkerCompletedEventHandler(CompletedHandler);
bgwExecute.ProgressChanged += new
ProgressChangedEventArgs(ProgressChanged);
bgwExecute.RunWorkerAsync();
}

```

If the UpdateDB method that is called by the BackgroundWorker component returns the value False, you need to display a message box to the user that indicates that the update failed. Which code segment should you use?

```

A. void WorkHandler(object sender, DoWorkEventArgs e) {
if (!UpdateDB()) MessageBox.Show("Update failed");
}
B. void CompletedHandler(object sender, RunWorkerCompletedEventArgs e) {
if (!UpdateDB()) MessageBox.Show("Update failed");
}
C. void WorkHandler(object sender, DoWorkEventArgs e) {
D. Result = UpdateDB();
}

```

```

void CompletedHandler(object sender, RunWorkerCompletedEventArgs e) {
    if (!(bool) e.Result) MessageBox.Show("Update failed");
}
E. void WorkHandler(object sender, DoWorkEventArgs e) {
F. Result = UpdateDB();
}
void CompletedHandler(object sender, RunWorkerCompletedEventArgs e) {

```

Answer: C

QUESTION: 76

You create an application that provide accessibility features. Your standard forms display a background image. When the user selects Use High Contrast in the Accessibility Options in Control Panel, you want this image to be removed. You need to add an event to handle this accessibility setting change. Which event should you use?

- A. this.StyleChanged
- B. SystemEvents.UserPreferenceChanged
- C. this.ChangeUICues
- D. SystemEvents.DisplaySettingsChanged

Answer: B

QUESTION: 77

You are creating a Windows Forms application. You add an ErrorProvider component named erpErrors and a DateTimePicker control named dtpStartDate to the application. The application also contains other controls.

You need to configure the application to display an error notification icon next to dtpStartDate when the user enters a date that is greater than today's date. Which two action should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. Ans For the Validating event of dtpStartDate, create an event handler named VerifyStartDate.
- B. For the Validated event of dtpStartDate, create an event handler named VerifyStartDate.
- C. In the Properties Window for dtpStartDate, set the value of Error on erpErrors to Date out of range .
- D. In VerifyStartDate, call erpErrors.SetError(dtpStartDate, "Date out of range") if the value of dtpStartDate value is greater than today's date.

E. Ans In VerifyStartDate, call `erpErrors.SetError(dtpStartDate, null)` if the `dtpStartDate.Value` is greater than today's date.

Answer:

Pending. Please send your suggestions to support@iqa.com

Download Full Version From <https://www.certkillers.net>



DON'T KNOW
OR NO PREFERENCE

Pass your exam at First Attempt....Guaranteed!